

The EON Model of Intervention Protocols and Guidelines

Samson W. Tu, M.S., Mark A. Musen, Ph.D.

Section on Medical Informatics, Stanford University School of Medicine
Stanford, CA 94305-5479

We present a computational model of treatment protocols abstracted from implemented systems that we have developed previously. In our framework, a protocol is modeled as a hierarchical plan where high-level protocol steps are decomposed into descriptions of more specific actions. The clinical algorithms embodied in a protocol are represented by procedures that encode the sequencing, looping, and synchronization of protocol steps. The representation allows concurrent and optional protocol steps. We define the semantics of a procedure in terms of an execution model that specifies how the procedure should be interpreted. We show that the model can be applied to an asthma guideline different from the protocols for which the model was originally constructed.

INTRODUCTION

The use of written guidelines and protocols¹ is becoming increasingly recognized as a method of assuring the quality of medical care and reducing the cost. A computer representation of these protocols and guidelines will facilitate their validation and dissemination, and it will allow the development of decision-support systems for protocol-based care that can consider patient-specific situations. Modeling clinical protocols and building protocol-based decision-support systems have been major themes of work in our laboratory. In the past 10 years, we have gained extensive experience modeling protocols in domains such as oncology [1], hypertension [2], and AIDS [3]. We have come to recognize the scope of the domain knowledge that we need to represent, the possible tasks in which a protocol may be used, and the alternative computational approaches that can be used to solve tasks related to protocol-based care. From our past modeling efforts, we have created a computable representation of treatment protocols called the EON protocol model. Creating this model has allowed us to develop decision-support systems that are reusable across different clinical domains [4] and to generate domain-specific editors for acquiring and maintaining protocol information [5]. We present our protocol representation here not because it is the definitive model that can be used for all possible protocols, but because it is a well-tested model that

has proved sufficient for representing a large class of treatment protocols, and that can serve as the basis for developing more general protocol representations.

AN EXAMPLE

The EON protocol model is best understood by means of an example. The model was originally developed for modeling clinical-trial protocols. To show that it can be used in more general settings, we will apply it to a guideline for managing severe adult asthma [6].

This guideline, published by the British Medical Association, divides potential cases of severe asthma into three classes: uncontrolled, acute severe, and life-threatening. We will focus on the treatment suggestions for acute severe asthma. After recognizing an acute severe case, the care provider is instructed to give oxygen 40–60% if available, then to administer (1) nebulized salbutamol 5 mg or nebulized terbutaline 10 mg and (2) oral prednisolone 30–60 mg or intravenous hydrocortisone 200 mg. Ten to twenty minutes after nebulization, if signs of acute asthma persist, then the care provider is instructed to arrange hospitalization and either (1) to repeat the nebulized β_2 agonist and to give nebulized ipratropium 500 micrograms, or (2) to give subcutaneous terbutaline, or (3) to give slow intravenous aminophylline 250 mg. The guideline also describes a follow-up plan that involves seeing the patient in the clinic, monitoring symptoms and peak flow, and setting up a self-management plan.

MODEL OF TREATMENT PROTOCOL

We consider treatment protocols as describing plans of actions. The instruction to give 5 mg of nebulized salbutamol, for example, specifies an action to be carried out. A protocol, such as the asthma guideline described above, specifies a set of actions and their sequencing relationships. It may not be completely deterministic, but may suggest only that one or some of the possible alternatives be given. A computable model of a treatment protocol therefore has a declarative and a procedural aspect. The declarative aspect defines parts of the protocol, their properties, and the relationships among them. The procedural aspect specifies the temporal sequencing, branching, and looping of prescribed or suggested treatment interventions. We model the declarative aspect of a protocol within an object-oriented

¹ For the purpose of this paper, the terms “guideline” and “protocol” will be used synonymously.

framework, where objects are instances of classes that are organized into an *is-a* hierarchy. We model the procedural aspect of a protocol by (1) defining a graph structure that represents the temporal relationships between parts of a protocol, and (2) describing an *execution model* of the graph structure. The execution model gives the graph structure operational semantics by describing how an application should interpret the structure as a plan of actions extended over time.

Protocol Representation

As is commonly done in the planning literature, we adopt the idea of hierarchical decomposition to manage potentially complex algorithms. Each *composite protocol* is recursively decomposed into a series of finer-granularity protocols until we reach the basic *elementary protocol* that cannot be decomposed any further. For each clinical domain that we model, we define classes of protocols that are relevant for the domain. Thus, if we were representing breast-cancer protocols, we would define protocols for carrying out chemotherapies, radiation treatments, and relevant monitoring activities. For the purpose of modeling asthma guidelines, such as the one sketched above, we define two classes of composite protocols (*clinical_protocol* representing top-level guidelines, and *regimen* representing treatment procedures for managing particular types of asthma) and two classes of elementary protocols (*drug_prescription* for specifying details of drug administrations, and *generic_protocol* which is used to model simple procedures not otherwise specified, such as administration of oxygen therapy). Instances of each class of protocol have a set of attributes that specify details of the instance. For example, instances of the *drug_prescription* protocol have attributes such as "maximum_dose," "minimum_dose," and "dose_unit." The example asthma guideline as a whole is modeled as an instance of *clinical_protocol*. The management of each type of asthma is modeled as a *regimen* protocol. Each regimen is decomposed into a set of *drug_prescriptions* and *generic_protocol* steps.

We define *procedures* to model the clinical algorithm underlying a composite protocol. A procedure is a plan consisting of the temporal sequencing of prescribed or suggested actions. We can visualize a procedure as a directed multigraph that has nodes of more than one class. The simplest type of procedure we have defined is a *conjunctive* procedure, which has two types of nodes, *start step* and *protocol step*, and uses directed arcs called *selections* to make choices among alternative protocol steps (Figure 1). A selection has a name and an associated selection condition. The nodes from which the arcs originate are the *predecessors* of the nodes to which the arcs point. Destination nodes of the arcs are the *successors* of the nodes originating the arcs.

Depending on the selection conditions, two or more protocol steps may be active at the same time.

A treatment protocol is applied to a particular case (subject or person) at a particular time. Thus, a computable model of a protocol must know how to reference information about the case, and have a language for describing relevant patient conditions. For example, in the procedure shown in Figure 1, the selection conditions include predicates that reference patient conditions such as "respirations ≥ 25 breaths/minute." We refer to the representation of patient data assumed by the protocol as the *case model* of the protocol. It is important because it is the basis of the patient-description language used in the protocols. In the EON model, we assume that a case is described by a collection of objects, where information about a case is indexed by a unique case identifier, and where the information about a case can be modeled as objects and their attributes and attribute values. An object may be an instance of, say, one laboratory-test result class and have attributes such as "name," "value," "unit," "upper_bound," "lower_bound," and so on. All patient data are considered to hold during a time interval (which may collapse into a time point), denoted by the start and stop times of the interval. At run time, patient data from an electronic patient record must be (1) loaded into a system that implements the EON model and (2) translated into the terms of the case model. Alternatively, expressions written in the EON patient-description language must be translated into queries understandable by the underlying database system.

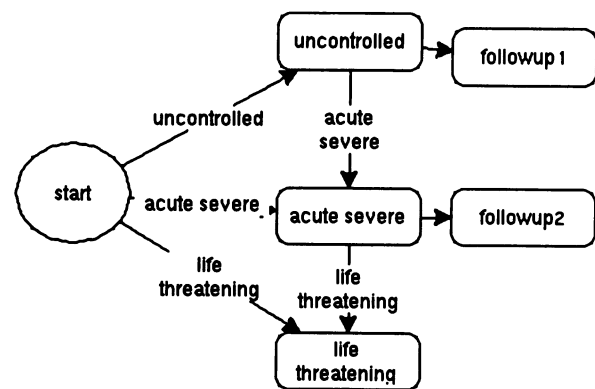


Figure 1. The algorithmic part of the asthma guideline is modeled using a procedure. The rounded rectangles represent protocol steps in the procedure. The directed arcs (selections) represent temporal precedence relations among protocol steps. The algorithm starts with three selections, where the selection conditions define what to look for in "uncontrolled," "acute severe," and "life-threatening" asthma.

One special class of objects in a case model is the *intervention* class. Interventions represent the

actions that have been taken by health-care providers to change the state of a patient. Thus, much of a treatment protocol specifies different kinds of interventions that should be carried out. Corresponding to the drug-prescription protocol that specifies how a drug should be given, the case model has a drug-administration intervention that describes how the drug is actually prescribed. The relationship between attributes of protocols and attributes of interventions is defined through *intervention rules*. These rules give meaning to the domain-specific attributes of the protocols. For example, in the example guideline described previously, the dose of the drug prednisolone administered to the patient should be between 30 and 60 mg, where the two bounds are modeled in EON as the minimum and maximum dose attributes in the prednisolone drug-prescription protocol. Because both interventions and protocols are modeled as objects, the same EON predicate language is used to specify relationships among these objects.

When a care provider performs an intervention such as prescribing a drug, she may alter some aspects of the intervention (e.g., changing the dose or suspending the prescription). We model such alterations as changes in the attributes or *state* of the intervention. In the EON model, an intervention may be in one of a limited number of states, depending on the class of the intervention. For example, for an intervention corresponding to a management regimen, we define three possible states: *active*, *completed*, and *aborted*. Possible transitions are limited to those between the active and completed and between the active and aborted states. For a drug-administration intervention, we may have an additional *suspended* state. We refer to instructions regarding how to change the attribute values and states of an intervention as *revision rules*. In the regimen for managing acute severe asthma, a revision rule may specify that, if, while being treated under this regimen, the patient's condition worsens from "acute severe" to "life-threatening," then the regimen for managing "acute severe" asthma should be aborted. In a different protocol, a revision rule may be much more complex, requiring, for example, a medication to be suspended until some condition holds, then to be resumed with attenuated dose for some specified interval before the full dose is given again.

The management regimen for acute severe asthma calls for initial treatment using oxygen (if it is available) and two drugs out of several possible alternatives, and then, depending on whether symptoms are resolved, additional drug treatment. The conjunctive procedure described earlier cannot express the notion of alternative actions, where the guideline gives no preference for one action over another. To increase the expressiveness of a procedure, we

introduce the notion of *junctions*, which extend the model of procedure by allowing different kinds of choices in the algorithm. There are several types of junctions that we have introduced. The decomposition of the acute severe asthma regimen uses *one-of* junctions (Figure 2). These junctions allow us to express in the EON model non-deterministic alternatives that occur frequently in clinical guidelines.

One-of junctions come in pairs: *one-of* and *end-one-of*. All paths starting from a *one-of* must end in the corresponding *end-one-of*. Only one of the selections starting from a *one-of* junction can be taken. Thus, nodes following a *one-of* denote disjoint alternatives for which the protocol specifies no explicit preference.

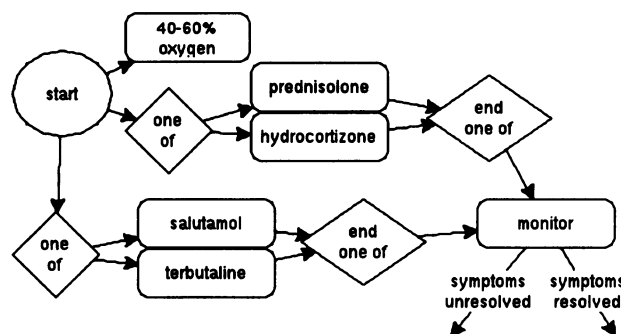


Figure 2. Part of the procedure decomposing the regimen for managing acute severe asthma. The last part of the regimen can be modeled using the same *one-of* construct.

Execution Model of a Procedure

To provide operational semantics for procedures used to model protocols and guidelines, we describe an execution model that indicates how the protocol steps, selections, and junctions should be interpreted. A patient being treated on the protocol has one or more active protocol-specified interventions. We keep track of where the patient is in the procedure by using a function *M* to map nodes of the procedure to 0 (inactive) or 1 (active).² A node of a procedure is activated if its marking changes from 0 to 1. For a node (that is not a start node) to become active, (1) the predecessors of the node must be active, (2) all interventions associated with protocol predecessors of the node must be completed or aborted, and (3) all selection conditions of the arcs from the predecessors must be *true*.

²Note that the binary states of a *node* in a procedure graph are distinct from the domain-specific states of the corresponding *intervention*.

The algorithm for executing a simplified conjunctive procedure is as follows:

1. If the procedure has not been activated before (i.e., $M(\text{node}) = 0$ for all nodes in the procedure), activate the *start* node. If the procedure has been activated before, go to 2.
2. Evaluate the revision rules of all active nodes, if there are any.
3. For each node that is a successor of an activated node, check to see if the node can be activated. If yes, then add the node to the list (L) of nodes to activate.
4. If L is empty, then exit.
5. Deactivate those nodes whose successors are in L.
6. Activate the nodes in L.
7. Go to 2.

To extend the execution model to cover a procedure with junctions, we consider a junction to be a node that is completed as soon as it is started. To keep track of junctions that may be nested within each other, the execution model maintains a stack of active junctions. Thus, upon encountering a *one-of* junction, the junction is pushed onto the stack. The junction is popped from the stack when the corresponding *end-one-of* junction becomes active. Those protocol steps that are active when the junction stack is non-empty specify interventions that are allowed, but not necessarily prescribed, by the protocol. In Figure 2, we use *one-of* junctions to model the guideline specification that *either prednisolone or hydrocortizone and either salutamol or terbutaline* be given to the patient.

The execution model thus provides an operational mechanism for specifying the meaning of procedures that we use to represent algorithms described in clinical guidelines. Describing the protocol steps in the form of a directed graph by itself does not tell a machine how to *interpret* the corresponding data structure. Determining the semantics of a graph can be a problem especially when protocol steps may be active concurrently. By describing a precise execution model, we make our procedural definition interpretable by a computer program in a precise manner.

DISCUSSION

The EON protocol model was the basis of the protocol-based decision-support module implemented for the T-HELPER system, a computer-based patient-record system designed to assist care providers in managing patients who are infected with the HIV virus. We implemented an interpreter for the protocols represented in the system and created mappings that translated patient data stored in a relational database into terms used by the protocol

case model. By considering a protocol as a skeletal plan to be refined, the decision-support program in T-HELPER generates drug-dosing and work-up suggestions based on the prescription of the clinical protocol [5].

In contrast to rule-based guideline representations, such as those written in the Arden Syntax, which emphasize creating individual units of medical knowledge and linking them together to represent a series of related decisions [7], we take the approach of developing a model of treatment protocols and representing different types of protocols and guidelines within that model. This protocol model, although complex, has proved to be highly adaptable for encoding different types of treatment protocols. In the T-HELPER project, it was used to represent protocols for treating patients infected with the HIV virus. We subsequently used the same model to represent an asthma guideline, which on the surface is very different from the clinical-trial protocols for which the model was initially constructed. In another recent experiment, by creating new domain-specific subclasses of protocols and modifying the vocabulary used in the system, we converted T-HELPER to a prototype system that generates protocol-based decision support in the domain of breast-cancer [8]. For that experiment, because we used the same protocol representation and case model, we were able to reuse the same interpreter and data mapping to generate advice based on a breast-cancer protocol.

This experiment applying the EON protocol model to a breast-cancer protocol shows that we can reuse the program code designed for managing AIDS patients to provide protocol-based decision support in oncology. The EON model also allowed us to generate, automatically, using PROTÉGÉ-II tools, domain-specific editors for acquiring and maintaining breast-cancer protocols [6]. The availability of such tools for AIDS protocols had proved invaluable during the development of the T-HELPER system.

A protocol representation that bears some similarity to ours is that of GEODE-CM [9]. In the GEODE-CM model, nodes in a finite-state machine represent patient states for which a guideline may specify one or more management actions. Transitions takes place when the system learns more information about the patient or when the patient's condition has changed substantially. It appears that this model has some advantages in representing diagnostic guidelines, where patient states are the primary objects of concern. However, the GEODE-CM framework lacks a clear execution model for specifying sequencing, looping, and synchronization of possibly multiple concurrent actions.

The protocol model closest to ours is that of the DILEMMA project, the European effort to create a

common protocol representation for a number of health-care institutions [10]. The DILEMMA model shares with EON a hierarchical decomposition structure, a distinction between the prescriptive specification (protocol) and the instantiated interventions (actions), and a similar formulation for changes in the status of actions as state transitions. The two models differ in that, in the DILEMMA model, sequencing of protocol steps is part of the transitions in a state machine, whereas the semantics of the protocol-decomposition plan are enforced by an explicit execution model in EON. The execution model, among other things, synchronizes the operation of concurrent protocol steps. Thus, the EON model specifies clearly that, when a protocol step has two or more predecessors that may be active concurrently, the step is not activated until all of its predecessors have completed their actions. In the DILEMMA model modeling such behavior is left to the protocol writers.

Our experience in the domains of AIDS, oncology, and asthma suggests that the EON protocol model provides a good foundation for representing protocols and guidelines. The model provides a source of ideas and worked out examples for any attempt to standardize guideline representation. Our current work centers on modularizing the organization of our protocol knowledge base and extending the model so that it can be used to represent both diagnostic guidelines and guidelines that are less specific in their treatment suggestions. We will modularize our knowledge base by partitioning the abstract protocol model, its specialization to specific clinical domains, and general medical knowledge into a hierarchy of domain theories. This reorganization will make our knowledge base more maintainable, without changing the EON protocol model. Our extensions of the protocol model to represent diagnostic and less-well-specified protocols will require us to represent the intentions and goals of the guidelines explicitly. This approach to protocol representation is an area of active investigation in our laboratory [11].

ACKNOWLEDGMENTS

This work has been supported in part by grants LM05708 and LM05304 from the National Library of Medicine, by grant CA65426 from the National Cancer Institute, and by contract N66001-94-D-6052 supported by the Advanced Research Projects Agency. Dr. Musen is the recipient of National Science Foundation Young Investigator Award IRI-9257578.

We are grateful to Amar Das, Doug Fridsma, John Gennari, Peter Johnson, and Yuval Shahar for ongoing discussions related to the automation of protocol-based care, and to Zaki Hasan for software development of the T-HELPER system.

REFERENCES

1. Tu, S.W., Kahn, M.G., Musen, M.A., et al. Episodic skeletal-plan refinement on temporal data. *Communications of the ACM*, 1989; 32:1439-55.
2. Musen, M.A. Automated support for building and extending expert models. *Machine Learning*, 1989; 4:349-77.
3. Musen, M.A., Carlson, C.W., Fagan, L.M., et al. T-HELPER: Automated support for community-based clinical research. *Proceedings of the Sixteenth Annual Symposium on Computer Applications in Medical Care*. Baltimore MD, 1992; 719-23.
4. Musen, M.A., Tu, S.W., Das, A.K., Shahar, Y., EON: A component-based approach to automation of protocol-directed therapy. *Journal of American Medical Informatics*, forthcoming.
5. Tu, S.W., Eriksson, H., Gennari, J., et al. Ontology-based configuration of problem-solving methods and generation of knowledge-acquisition tools: Application of PROTÉGÉ-II to protocol-based decision support. *Artificial Intelligence in Medicine*, 1995; 7:257-89.
6. Prasad, A.B., ed. *British National Formulary, Number 30*. London: The Pharmaceutical Press, 1995; 117.
7. Sherman, E.H., Hripcsak, G., Staren, J., et al. Using intermediate states to improve the ability of the Arden Syntax to implement care plans and reuse knowledge. *Proceedings of the Nineteenth Annual Symposium on Computer Applications in Medical Care*. New Orleans, 1995; 238-42.
8. Musen, M.A., Tu, S.W., Hasan, Z. Reuse of the EON architecture to automate protocol-based care: From AIDS to Breast Cancer. 1996 AMIA Fall Symposium (Abstract).
9. Stoufflet, P.E., Deibel, S.R.A., Traum, J.H., et al. A state-transition method of modeling clinical encounters. *Proceedings of the AMIA Spring Congress*. 1995;81 (Abstract).
10. Herbet, S.I. Informatics for care protocols and guidelines: Towards a European knowledge model. In C. Gordon and J.P. Christensen eds. *Health Telematics for Clinical Guidelines and Protocols*, IOS Press, 1995; 257-89.
11. Shahar, Y., Miksch, S., and Johnson, P.D. An intention-based language for sharing clinical guidelines. *Proceedings of the 1996 AMIA Fall Symposium*. Washington DC, 1996; this volume.